Solving L_0-Problems via Mixed-Integer Optimization

Théo Guyard

Inria, Centre de l'Univsersité de Rennes, France

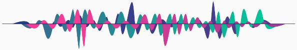
Insa Rennes, IRMAR CNRS UMR 6625, France

LS2N seminar
7th of March, 2024
Nantes, France

Sparse Optimization

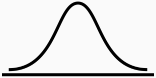
Two goals, one problem

Minimize a function $\xleftrightarrow{\textbf{Sparse optimization}}$ **Sparse solution**



Signal processing

Machine learning



High-dim. statistics

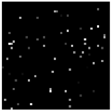
And many others

Signal processing

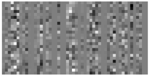
Compressive sensing

Linear measurement through $A \in \mathbb{R}^{m \times n}$

Sparse signal
$x \in \mathbb{R}^n$



Observation
$y \in \mathbb{R}^m$



Recover x from (A, y)

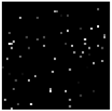
Find x **such that** $y \simeq Ax$

Signal processing

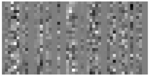
Compressive sensing

Linear measurement through $A \in \mathbb{R}^{m \times n}$

Sparse signal
$x \in \mathbb{R}^n$



Observation
$y \in \mathbb{R}^m$



Recover x from (A, y)

Find x **such that** $y \simeq Ax$

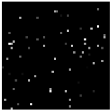
$m \ll n$: no unique solution

Signal processing

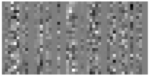
Compressive sensing

Linear measurement through $A \in \mathbb{R}^{m \times n}$

Sparse signal
$x \in \mathbb{R}^n$



Observation
$y \in \mathbb{R}^m$



Recover x from (A, y)

Find x **sparse such that** $y \simeq Ax$

$m \ll n$: no unique solution

High-dimensional statistics

Sparse GLM

Features $\mathrm{A} \in \mathrm{R}^{m \times n}$

Targets $\mathrm{y} \in \mathrm{R}^{m}$

↓

$\max_{\mathrm{x}} \; \mathcal{L}(\mathrm{Ax}, \mathrm{y})$

No unique solution when $m \ll n$

↓

$\max_{\mathrm{x}} \; \mathcal{L}(\mathrm{Ax}, \mathrm{y})$ with x sparse

High-dimensional statistics

Sparse GLM	**Sparse PCA**
Features $A \in \mathbb{R}^{m \times n}$	Features $A \in \mathbb{R}^{m \times n}$
Targets $y \in \mathbb{R}^{m}$	Covariance $\Sigma = A^{\mathrm{T}}A$
↓	↓
$\max_{x} \; \mathcal{L}(Ax, y)$	$\max_{\lVert x \rVert_2 = 1} \; x^{\mathrm{T}}\Sigma x$
No unique solution when $m \ll n$	Not relevant when $m \ll n$
↓	↓
$\max_{x} \; \mathcal{L}(Ax, y)$ with x sparse	$\max_{\lVert x \rVert_2 = 1} \; x^{\mathrm{T}}\Sigma x$ with x sparse

Machine learning

Heart disease dataset (LIBSVM)

Age	Sex	Cholesterol	Blood pressure	...	**Disease**
31	M	50.3 mg/dl	95 mm/hg	...	**No**
35	F	54.9 mg/dl	98 mm/hg	...	**Yes**
42	F	49.8 mg/dl	92 mm/hg	...	**Yes**
37	M	59.1 mg/dl	89 mm/hg	...	**No**
...	...	...	...	...	...

Machine learning

Heart disease dataset (LIBSVM)

Age	Sex	Cholesterol	Blood pressure	...	**Disease**
31	M	50.3 mg/dl	95 mm/hg	...	**No**
35	F	54.9 mg/dl	98 mm/hg	...	**Yes**
42	F	49.8 mg/dl	92 mm/hg	...	**Yes**
37	M	59.1 mg/dl	89 mm/hg	...	**No**
...	...	...	...	...	...

Data $\longrightarrow$ **Logistic regression** $\longrightarrow$ **Estimator** x

Machine learning

Heart disease dataset (LIBSVM)

Age	Sex	Cholesterol	Blood pressure	...	**Disease**
31	M	50.3 mg/dl	95 mm/hg	...	**No**
35	F	54.9 mg/dl	98 mm/hg	...	**Yes**
42	F	49.8 mg/dl	92 mm/hg	...	**Yes**
37	M	59.1 mg/dl	89 mm/hg	...	**No**
...	...	...	...	...	...

Data $\longrightarrow$ **Logistic regression** $\longrightarrow$ **Estimator** x

Involves all the features

Explainability and robustness issues

Machine learning

Heart disease dataset (LIBSVM)

Age	Sex	Cholesterol	Blood pressure	...	**Disease**
31	M	50.3 mg/dl	95 mm/hg	...	**No**
35	F	54.9 mg/dl	98 mm/hg	...	**Yes**
42	F	49.8 mg/dl	92 mm/hg	...	**Yes**
37	M	59.1 mg/dl	89 mm/hg	...	**No**
...	...	...	...	...	...

Data $\longrightarrow$ **Logistic regression** $\longrightarrow$ **Estimator** x

Force the use of only few features via sparsity in x

Involves all the features
Explainability and robustness issues

Objective, constraint or both ?

Minimize a function $\longleftrightarrow$ **Sparse optimization** $\longleftrightarrow$ **Sparse solution**

Loss function $F(x)$

Counting function $\|x\|_0$

Objective, constraint or both ?

Minimize a function $\longleftrightarrow$ **Sparse optimization** $\longleftrightarrow$ **Sparse solution**

Loss function $F(\mathrm{x})$

Counting function $\|\mathrm{x}\|_0$

Constrainted version

$$\begin{cases} \min_{\mathrm{x}} & F(\mathrm{x}) \\ \text{s.t.} & \|\mathrm{x}\|_0 \leq s \end{cases}$$

Minimized version

$$\begin{cases} \min_{\mathrm{x}} & \|\mathrm{x}\|_0 \\ \text{s.t.} & F(\mathrm{x}) \leq \epsilon \end{cases}$$

Penalized version

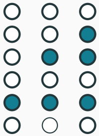
$$\min_{\mathrm{x}} F(\mathrm{x}) + \lambda\|\mathrm{x}\|_0$$

A bit of history

1990

Sparse
heuristics

MP, OMP,
IHT and co.

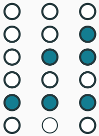


A bit of history

1990 | **2000**

Sparse heuristics

MP, OMP,
IHT and co.



Formulation with ℓ_0-norm

RIP, NSP,
EkR and co.

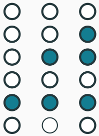
OMP
|||
ℓ_0-prob.

A bit of history

1990

Sparse heuristics

MP, OMP, IHT and co.



2000

Formulation with ℓ_0-norm

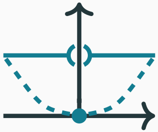
RIP, NSP, EkR and co.

OMP
|||
ℓ_0-prob.

2005

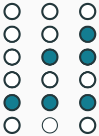
Convex approx. of ℓ_0-norm

Lasso, Elastic-Net and co.

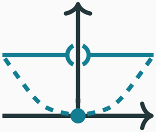


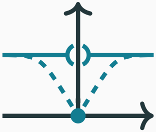
A bit of history

1990	2000	2005	2013
Sparse heuristics	Formulation with ℓ_0-norm	Convex approx. of ℓ_0-norm	Concave approx. of ℓ_0-norm
MP, OMP, IHT and co.	RIP, NSP, EkR and co.	Lasso, Elastic-Net and co.	SCAD, MCP CEL0 and co.



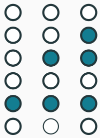
OMP $\equiv$ ℓ_0-prob.



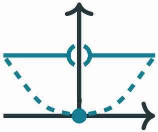


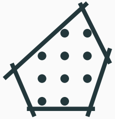
A bit of history

1990	2000	2005	2013	2016
Sparse heuristics	Formulation with ℓ_0-norm	Convex approx. of ℓ_0-norm	Concave approx. of ℓ_0-norm	Exact methods for ℓ_0-norm
MP, OMP, IHT and co.	RIP, NSP, EkR and co.	Lasso, Elastic-Net and co.	SCAD, MCP CEL0 and co.	MIP, BnB CP and co.



OMP
|||
ℓ_0-prob.





Mixed-Integer Optimization

Handeling the L0-norm with MIO tools

Problem

$\min_x F(x) + \lambda \|x\|_0$

$\downarrow$ Solver

Solution

Handeling the L0-norm with MIO tools

Problem

$\min_x F(x) + \lambda \|x\|_0$

Solver

Solution

Non-linear, non-convex,
non-smooth, NP-hard, ...

😵

Handeling the L0-norm with MIO tools

Problem

$\min_x F(x) + \lambda \|x\|_0$

Solver

Solution

Non-linear, non-convex,
non-smooth, NP-hard, ...

The ℓ_0-norm counts the number of non-zeros in vector x

It sums the entries of the binary vector z satisfying some logical relation with x

We have tools to deal with such binary vectors in MIO !

Fitting the MIO formalism

Linearizing the ℓ_0-norm

Real vector $x \in \mathbb{R}^n$ and binary vector $z \in \mathbb{B}^n$:

$$\|x\|_0 = 1^{\mathrm{T}} z \quad \text{if} \quad x \odot (1 - z) = 0$$

Fitting the MIO formalism

Linearizing the ℓ_0-norm

Real vector $x \in R^n$ and binary vector $z \in B^n$:

$$\|x\|_0 = 1^{\mathrm{T}} z \quad \text{if} \quad x \odot (1 - z) = 0$$

$$\boxed{\min_x F(x) + \lambda \|x\|_0}$$

Fitting the MIO formalism

Linearizing the ℓ_0-norm

Real vector $x \in \mathbb{R}^n$ and binary vector $z \in \mathbb{B}^n$:

$$\|x\|_0 = 1^{\mathrm{T}} z \quad \text{if} \quad x \odot (1 - z) = 0$$

$$\boxed{\min_x F(x) + \lambda \|x\|_0}$$

$$\downarrow$$

$$\boxed{\min_{x,z} F(x) + \lambda 1^{\mathrm{T}} z + H(x, z)}$$

Fitting the MIO formalism

Linearizing the ℓ_0-norm

Real vector $x \in \mathrm{R}^n$ and binary vector $z \in \mathrm{B}^n$:

$$\|x\|_0 = 1^{\mathrm{T}} z \quad \text{if} \quad x \odot (1 - z) = 0$$

$$\boxed{\min_x F(x) + \lambda \|x\|_0}$$

$$\downarrow$$

$$\boxed{\min_{x,z} F(x) + \lambda 1^{\mathrm{T}} z + H(x, z)}$$

$$\downarrow$$

$$\boxed{\begin{aligned} \min_{x,z} \quad & F(x) + \lambda 1^{\mathrm{T}} z \\ \text{s.t.} \quad & -Mz \le x \le Mz \end{aligned}}$$

Fitting the MIO formalism

Linearizing the ℓ_0-norm

Real vector $x \in \mathbb{R}^n$ and binary vector $z \in \mathbb{B}^n$:

$$\|x\|_0 = \mathbf{1}^{\mathrm{T}} z \quad \text{if} \quad x \odot (\mathbf{1} - z) = 0$$

$$\boxed{\min_x F(x) + \lambda \|x\|_0}$$

$$\downarrow$$

$$\boxed{\min_{x,z} F(x) + \lambda \mathbf{1}^{\mathrm{T}} z + H(x,z)}$$

$$\downarrow$$

$$\boxed{\begin{array}{rl} \min_{x,z} & F(x) + \lambda \mathbf{1}^{\mathrm{T}} z \\ \text{s.t.} & -Mz \leq x \leq Mz \end{array}}$$

Generic MIO solvers (Cplex, Gurobi, ...)
✗ Slow ✓ Generic w.r.t F/H

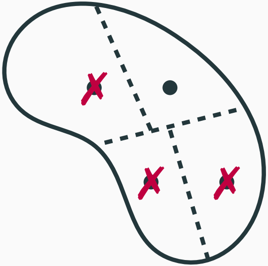
Specialized solvers (BnB, CP, ...)
✓ Fast ✗ Resticted to some F/H

Specialized Solution Methods

Branch-and-Bound algorithms

Branch-and-Bound

"Enumerate all candidate solutions and discard sub-optimal ones."



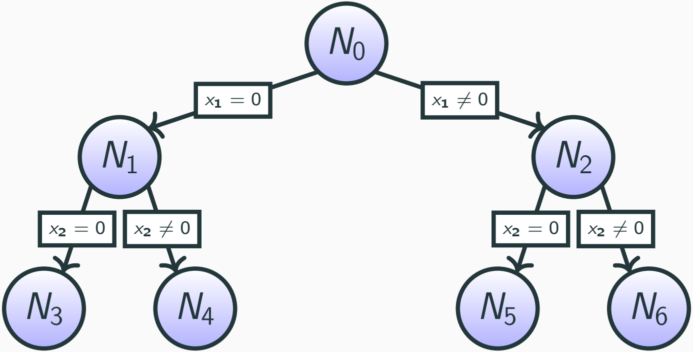
Main principles

Branching: Divide the search space

Bounding: Test whether a region can contain optimal solutions

Pruning: Discard regions without optimal solutions

Tree exploration



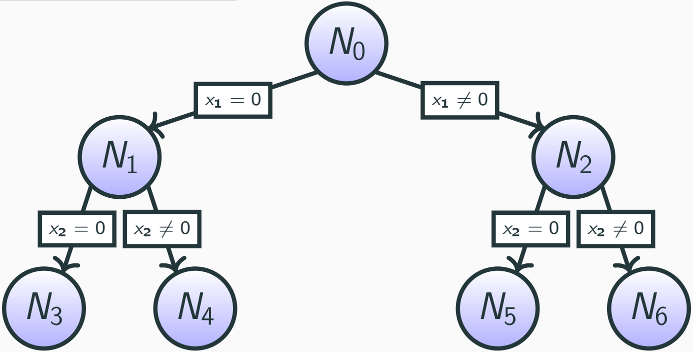
Tree exploration

Observation

If support of x is fixed, then

$$\min_{x} F(x) + \lambda\|x\|_0 + H(x)$$

is easy to solve.



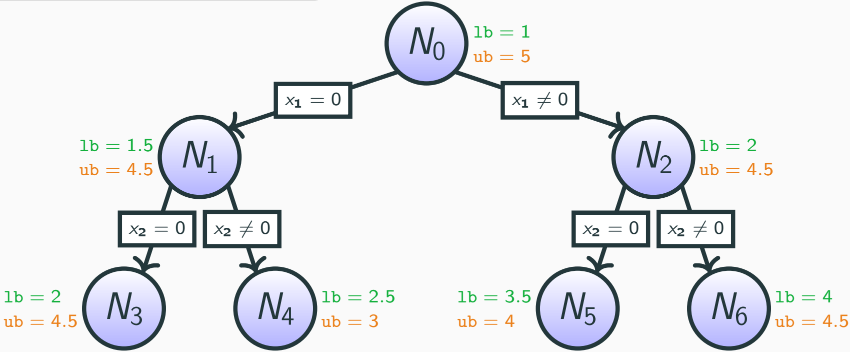
Tree exploration

Observation

If support of x is fixed, then

$$\min_{\mathbf{x}} F(\mathbf{x}) + \lambda\|\mathbf{x}\|_0 + H(\mathbf{x})$$

is easy to solve.



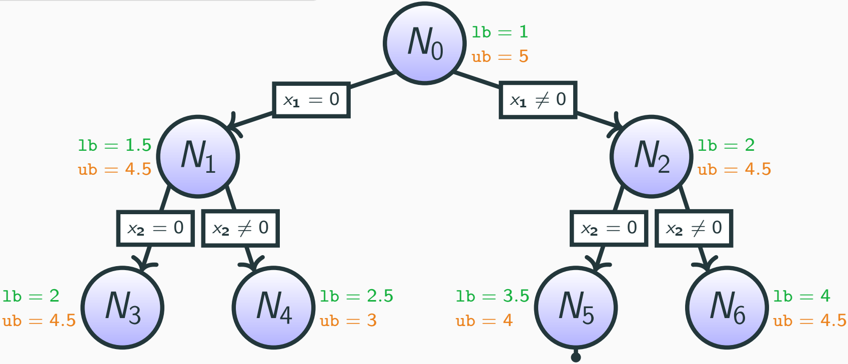
Tree exploration

Observation

If support of x is fixed, then

$$\min_{\mathrm{x}} F(\mathrm{x}) + \lambda\|\mathrm{x}\|_0 + H(\mathrm{x})$$

is easy to solve.



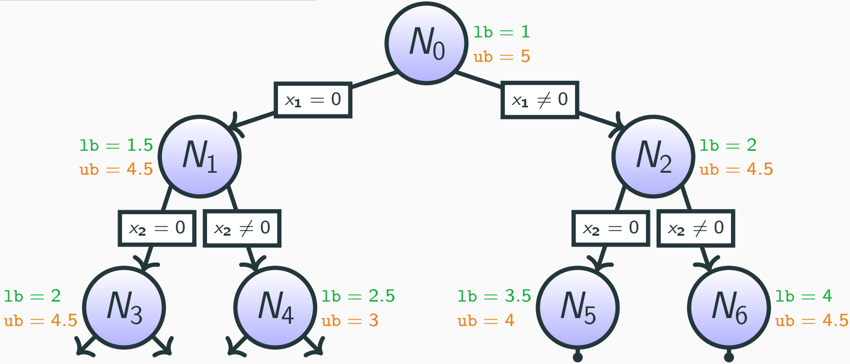
Tree exploration

Observation

If support of x is fixed, then

$$\min_x F(x) + \lambda\|x\|_0 + H(x)$$

is easy to solve.



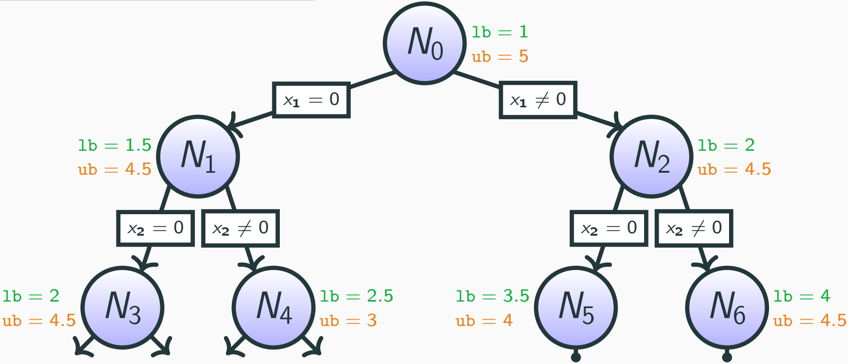
Tree exploration

Observation

If support of x is fixed, then

$$\min_{x} F(x) + \lambda\|x\|_0 + H(x)$$

is easy to solve.



All nodes explored or pruned $\longrightarrow$ Problem solved

Node processing

$(\mathcal{S}_0, \mathcal{S}_1)$

Node problem

The problem at node $\nu = (\mathcal{S}_0, \mathcal{S}_1)$ where $\mathcal{S}_0$ and $\mathcal{S}_1$ are the indices of x fixed to zero and non-zero reads

$$p^{\nu} = \begin{cases} \min_{\mathbf{x}} & F(\mathbf{x}) + \lambda\|\mathbf{x}\|_0 + H(\mathbf{x}) \\ \text{s.t.} & \mathbf{x}_{\mathcal{S}_0} = 0,\ \mathbf{x}_{\mathcal{S}_1} \neq 0 \end{cases}$$

Node processing

$(\mathcal{S}_0, \mathcal{S}_1)$

Node problem

The problem at node $\nu = (\mathcal{S}_0, \mathcal{S}_1)$ where $\mathcal{S}_0$ and $\mathcal{S}_1$ are the indices of x fixed to zero and non-zero reads

$$p^\nu = \begin{cases} \min_{\mathrm{x}} & F(\mathrm{x}) + \lambda\|\mathrm{x}\|_0 + H(\mathrm{x}) \\ \text{s.t.} & \mathrm{x}_{\mathcal{S}_0} = 0,\ \mathrm{x}_{\mathcal{S}_1} \neq 0 \end{cases}$$

Task: Find lower and upper bounds on p^ν that are **tight** and **tractable to compute**

Node processing

$(\mathcal{S}_0, \mathcal{S}_1)$

Node problem

The problem at node $\nu = (\mathcal{S}_0, \mathcal{S}_1)$ where $\mathcal{S}_0$ and $\mathcal{S}_1$ are the indices of x fixed to zero and non-zero reads

$$p^\nu = \begin{cases} \min_{\mathrm{x}} & F(\mathrm{x}) + \lambda\|\mathrm{x}\|_0 + H(\mathrm{x}) \\ \text{s.t.} & \mathrm{x}_{\mathcal{S}_0} = 0,\ \mathrm{x}_{\mathcal{S}_1} \neq 0 \end{cases}$$

Task: Find lower and upper bounds on p^ν that are **tight** and **tractable to compute**

Upper bounding

- We just need a feasible solution
- Fix entries of x that are still free to zero
- Optimize the resulting problem

Node processing

$(\mathcal{S}_0, \mathcal{S}_1)$

Node problem

The problem at node $\nu = (\mathcal{S}_0, \mathcal{S}_1)$ where $\mathcal{S}_0$ and $\mathcal{S}_1$ are the indices of x fixed to zero and non-zero reads

$$p^\nu = \begin{cases} \min_{\mathrm{x}} & F(\mathrm{x}) + \lambda\|\mathrm{x}\|_0 + H(\mathrm{x}) \\ \text{s.t.} & \mathrm{x}_{\mathcal{S}_0} = 0,\ \mathrm{x}_{\mathcal{S}_1} \neq 0 \end{cases}$$

Task: Find lower and upper bounds on p^ν that are **tight** and **tractable to compute**

Upper bounding

- We just need a feasible solution
- Fix entries of x that are still free to zero
- Optimize the resulting problem

Easy to solve

Upper-bounding problem

$$\min_{\mathrm{x}}\ F(\mathrm{x}_{\mathcal{S}_1}) + \lambda|\mathcal{S}_1| + H(\mathrm{x}_{\mathcal{S}_1})$$

Lower bounding

Idea: Convexify a part of the objective function

Node problem

$$p^{\nu} = \begin{cases} \min_{\mathrm{x}} & F(\mathrm{x}) + \lambda\|\mathrm{x}\|_0 + H(\mathrm{x}) \\ \text{s.t.} & \mathrm{x}_{\mathcal{S}_0} = 0,\ \mathrm{x}_{\mathcal{S}_1} \neq 0 \end{cases}$$

$\min_{\mathrm{x}} F(\mathrm{x}) + G(\mathrm{x})$

Lower bounding

Idea: Convexify a part of the objective function

Node problem

$$p^\nu = \begin{cases} \min_x & F(x) + \lambda\|x\|_0 + H(x) \\ \text{s.t.} & x_{\mathcal{S}_0} = 0,\ x_{\mathcal{S}_1} \neq 0 \end{cases}$$

$\min_x F(x) + G(x)$

$\boxed{\text{Bi-conjugacy } G^{\star\star}(x) \leq G(x)}$

Lower-bounding problem

$$\min_x F(x) + G^{\star\star}(x)$$

Lower bounding

Idea: Convexify a part of the objective function

Node problem

$$p^\nu = \begin{cases} \min_x & F(x) + \lambda\|x\|_0 + H(x) \\ \text{s.t.} & x_{\mathcal{S}_0} = 0,\ x_{\mathcal{S}_1} \neq 0 \end{cases}$$

$\min_x F(x) + G(x)$

$\boxed{\text{Bi-conjugacy } G^{\star\star}(x) \leq G(x)}$

Lower-bounding problem

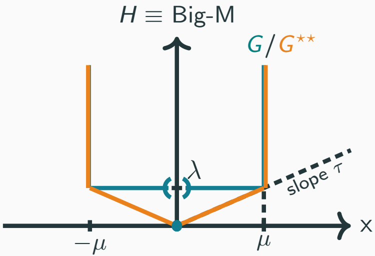
$$\min_x F(x) + G^{\star\star}(x)$$

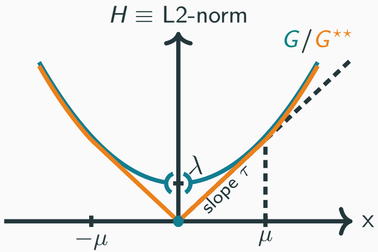
Closed-form expression

🥳

H proper, closed, coercive continuous at its minimum

Graphical intuition





Bi-conjugate closed-form

$$G^{\star\star}(x) = \begin{cases} \tau|x| & \text{if } |x| \leq \mu \\ G(x) & \text{otherwise} \end{cases}$$

Let's sum up !

ℓ_0-penalized problem

$$\min_x \; F(x) + \lambda \|x\|_0 + H(x)$$

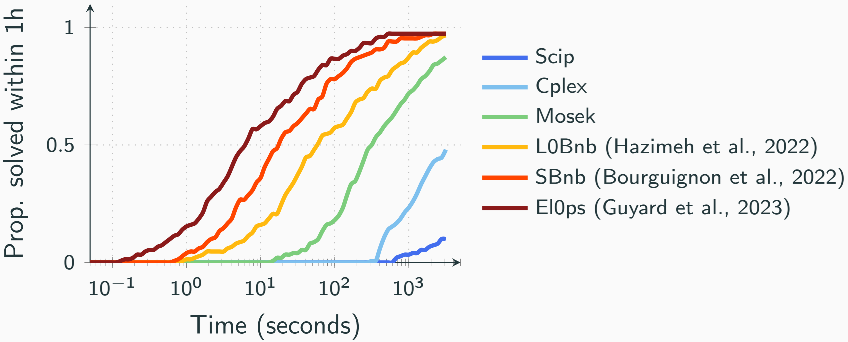
- MIP formalism
  - Linearize the ℓ_0-norm with a binary variable
  - Big-M strategy
- Generic solvers
  - Easy solution to implement
  - Unable to exploit sparsity
  - Numerically inefficient
- Specialized Branch-and-Bound
  - Tree exploration
  - Branch by fixing support of x
  - Compute upper and lower bounds at each node
  - Leverage bi-conjugacy to compute lower bounds

Overview of Numerical Performances

Overview of numerical performances

$$\min_x \; F(x) + \lambda\|x\|_0 + H(x)$$

Dataset : Sparse regression
F(·) : Least-squares loss
H(·) : Big-M constraints
λ : Set statistically



Overview of numerical performances

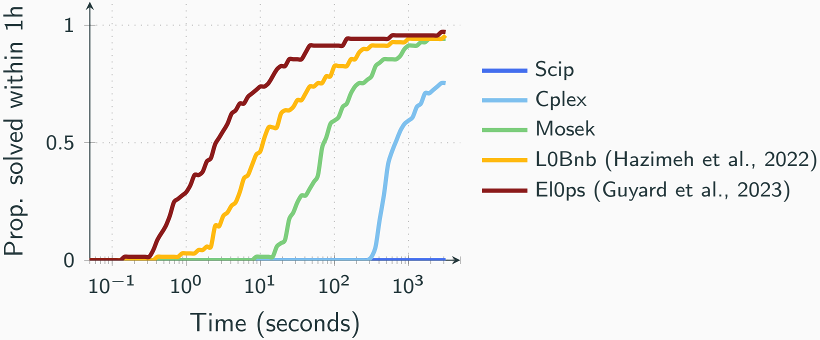
$$\min_x \; F(x) + \lambda\|x\|_0 + H(x)$$

Dataset : Sparse regression

F(·) : Least-squares loss

H(·) : L2-norm

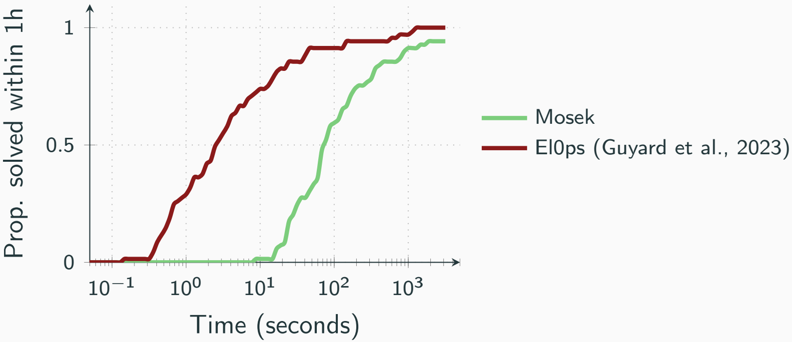
λ : Set statistically



Overview of numerical performances

$$\min_x \; F(x) + \lambda\|x\|_0 + H(x)$$

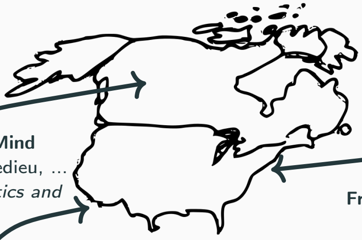
Dataset : Sparse classification
F(·) : Logistic loss
H(·) : L1-norm
$\boldsymbol{\lambda}$: Set statistically



Ongoing Research Directions

Contributors and research works

Non-exhaustive list



MIT
D. Bertsimas, R. Mazmuder, ...
MIO tools for ℓ_0-problems

Lund University
M. Carlsson, C. Olsson...
Quadratic envelope

Google Deep Mind
H. Hazimeh, A. Dedieu, ...
MIO-based heuristics and softwares

Frankfurt / Wurzburg Universities
C. Kanzow, A. Tillmann, ...
Optimality conditions

Berkley
A. Atamtürk, A. Gomès, ...
Convex-based acceleration

London Business School
J. Pauphilet, R. Cory-Wright, ...
Healthcare applications

Ponts ParisTech
M. De Lara, P. Chancelier, A. Parmentier, ...
Non-convex analysis for ℓ_0-norm, ML appli.

Centrale Nantes / ENSTA Bretagne
S. Bourguignon, J. Ninin, ...
Branch-and-Bound for ℓ_0-problems

Inria / CentraleSupélec
C. Herzet, C. Elvira, A. Arslan, ...
Generalization, acceleration

IRIT / I3S
E. Soubies, L. Blanc-Féraud, ...
Strong relax. of ℓ_0-norm

Take-home message

- In some cases, solving ℓ_0-norm problems exactly worths-it
- There exists Mixed-Integer Optimization tools to do so
- Structure exploitation is the key to achieve competitive performances
- Active research area

  $\rightarrow$ Theoretical results

  $\rightarrow$ Efficiency, flexibility and accessibility of solution methods

  $\rightarrow$ Software development

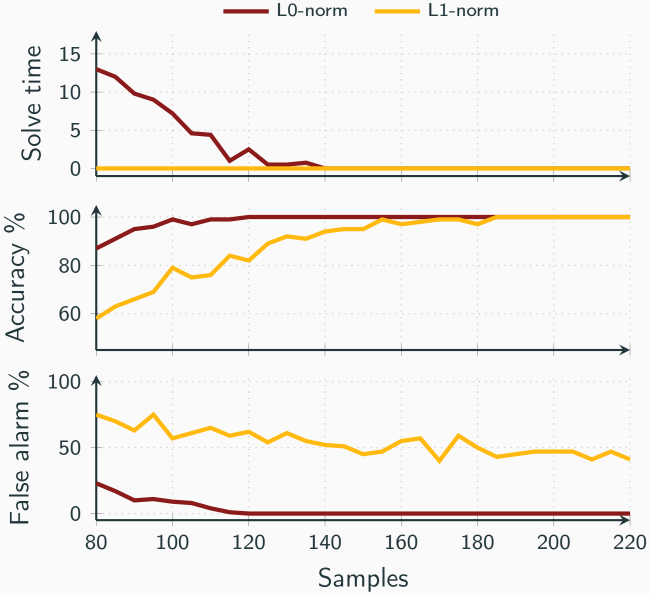
  $\rightarrow$ Diffusion to other communities

Question time

🤓

Supplementary Slides

Why solving L0 problems ?



Sparse regression

$y = Ax^{\dagger} + \epsilon$

2.000 features

10 non-zeros in $x^{\dagger}$

20dB noise

D. Bertsimas and B. Van Parys, Sparse High-Dimensional Regression: Exact Scalable Algorithms and Phase Transitions